Hackathon: The development and test of Java agent plugin

Zhang Xin, Apache Skywalking PMC

2019/5/11

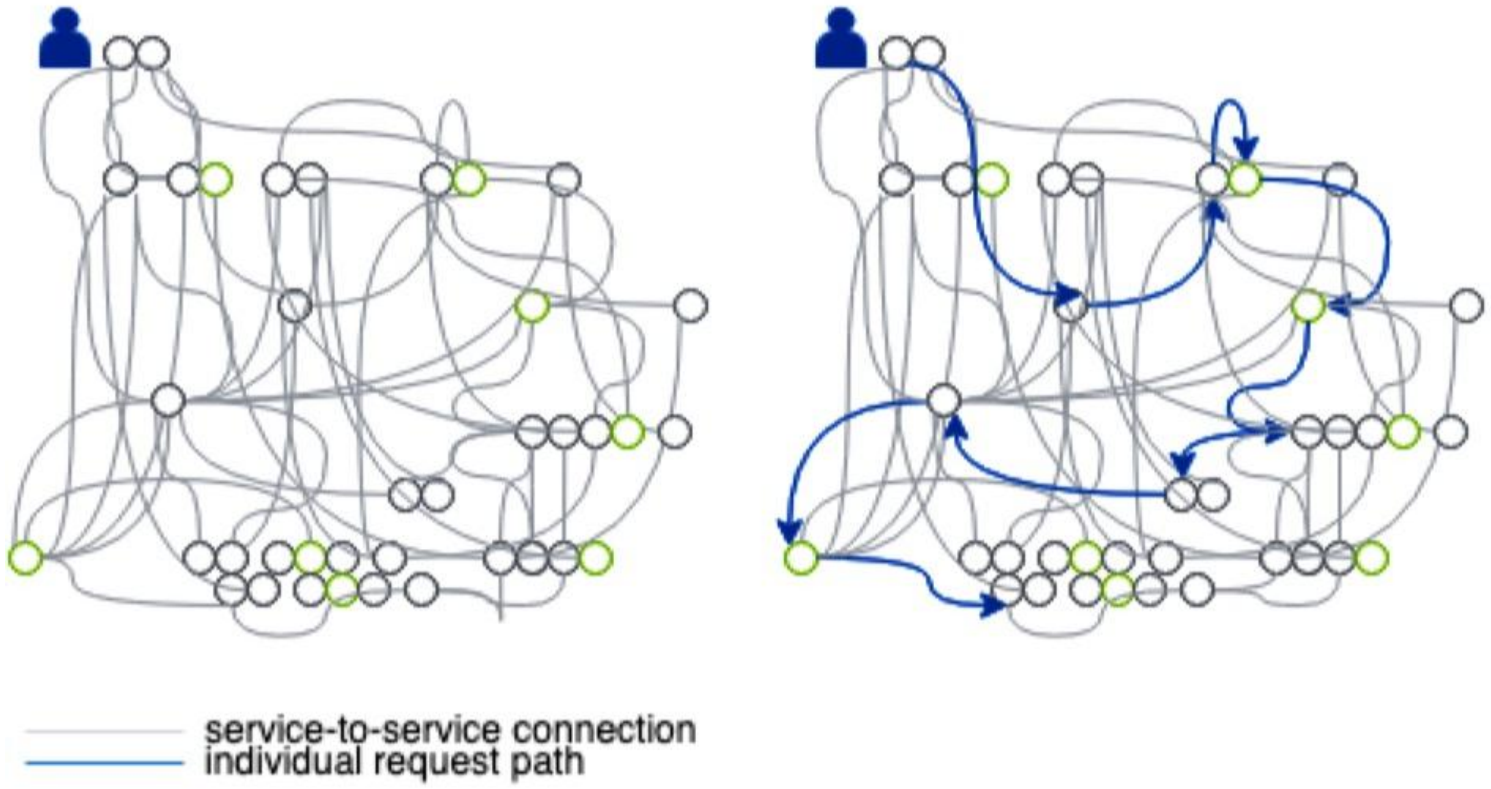
ABOUT ME

Zhang Xin(张鑫)

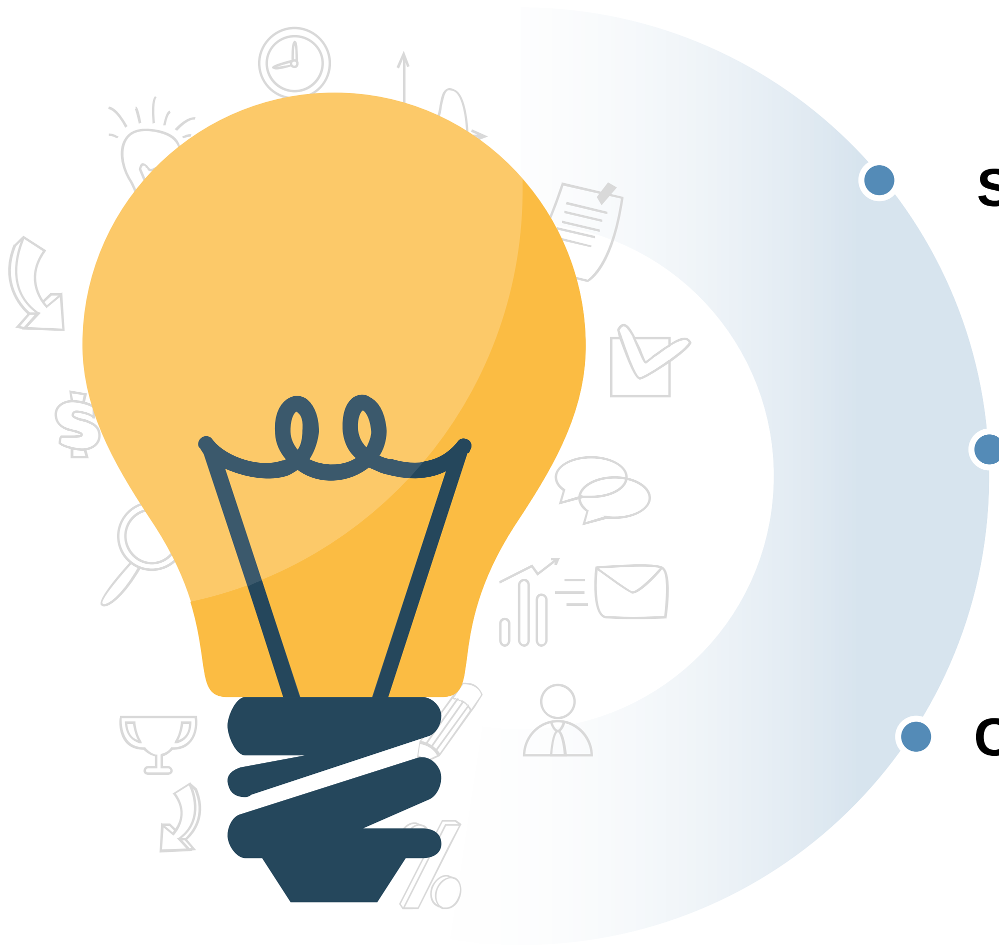
- Apache SkyWalking PMC
- Skywalking Node.js lead
- Github: https://github.com/ascrutae

CONTENTS

WHAT’S THE SKYWALKING



CONCEPT



Service	Instance
Trace Segment	Span
Context Carrier	Context Snapshot

LET’S CODE

Plugin Project Structure

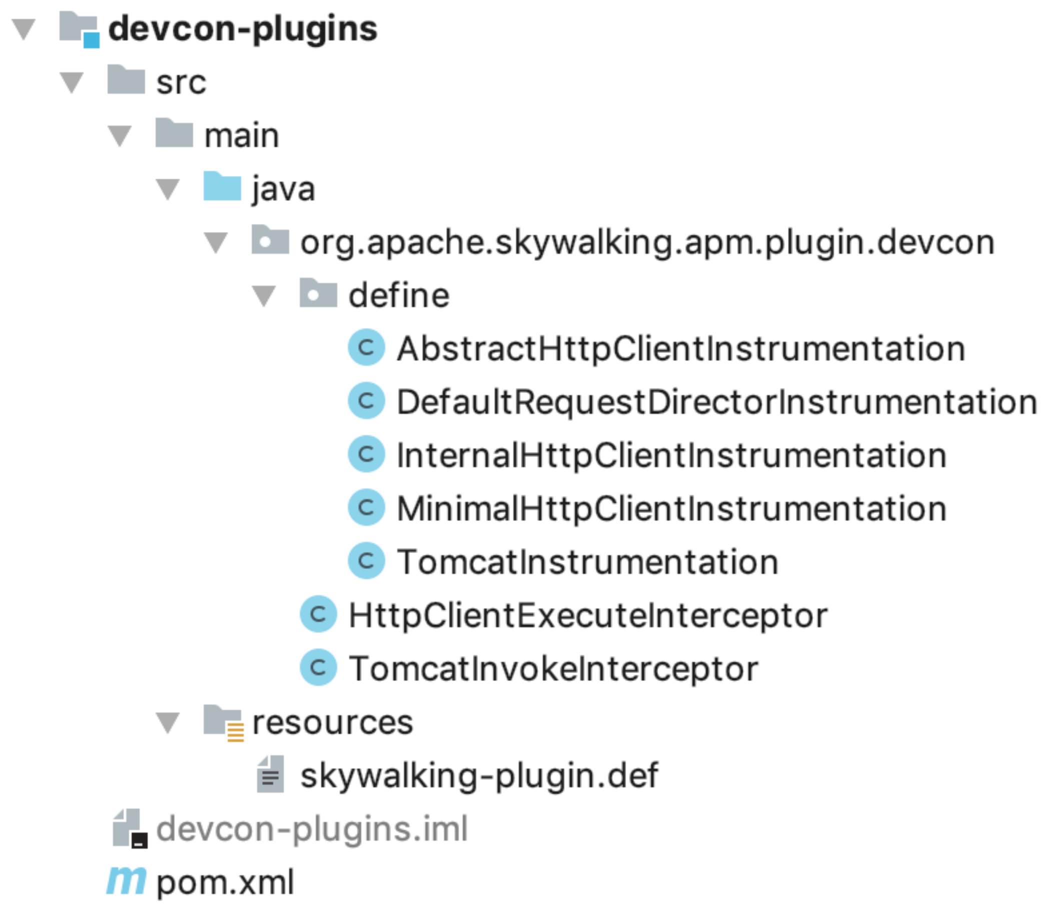
- skywalking-plugin.def
- instrumentations + interceptor

Steps to write plugin

- Find instrumentations
- Write plugin by right API

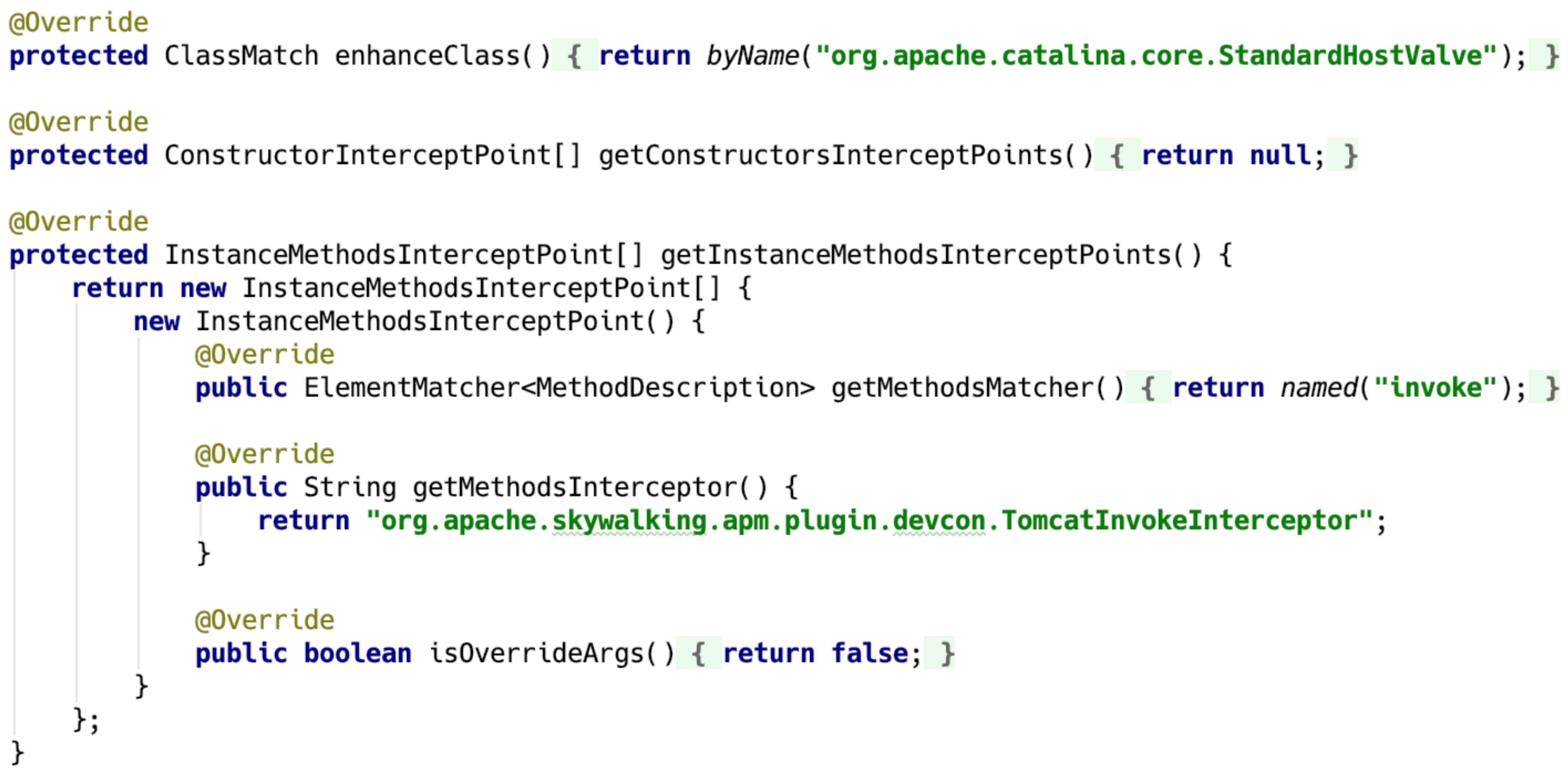
PLUGIN PROJECT

- skywalking-plugin.def
- Instrumentations
  - ClassInstanceMethodsEnhancePluginDefine
  - ClassStaticMethodsEnhancePluginDefine
- Interceptor
  - InstanceConstructorInterceptor
  - InstanceMethodsAroundInterceptor
  - StaticMethodsAroundInterceptor



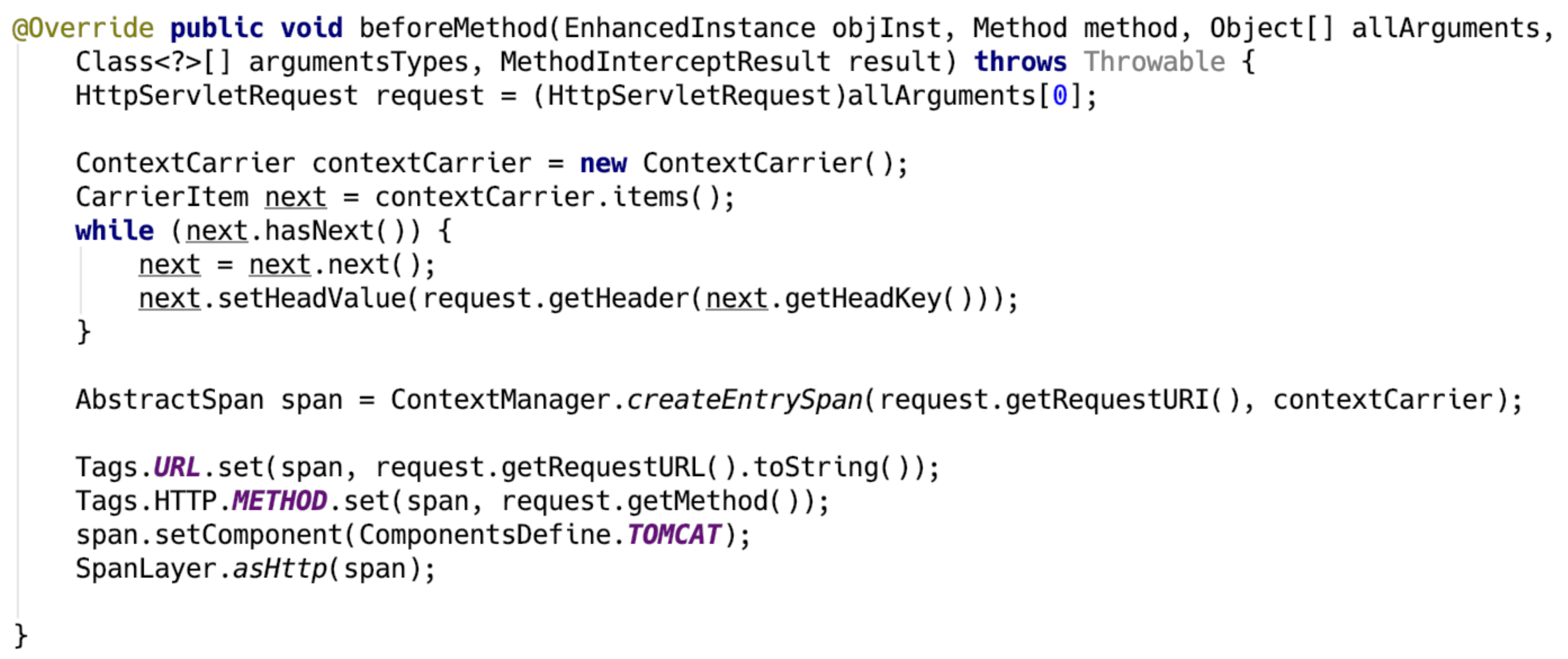
```
devcon-plugins
  src
    main
      java
        org.apache.skywalking.apm.plugin.devcon
          define
            AbstractHttpClientInstrumentation
            DefaultRequestDirectorInstrumentation
            InternalHttpClientInstrumentation
            MinimalHttpClientInstrumentation
            TomcatInstrumentation
          HttpClientExecuteInterceptor
          TomcatInvokeInterceptor
      resources
        skywalking-plugin.def
  devcon-plugins.iml
  pom.xml
```

THE TOMCAT PLUGIN-INSTRUMENTATION



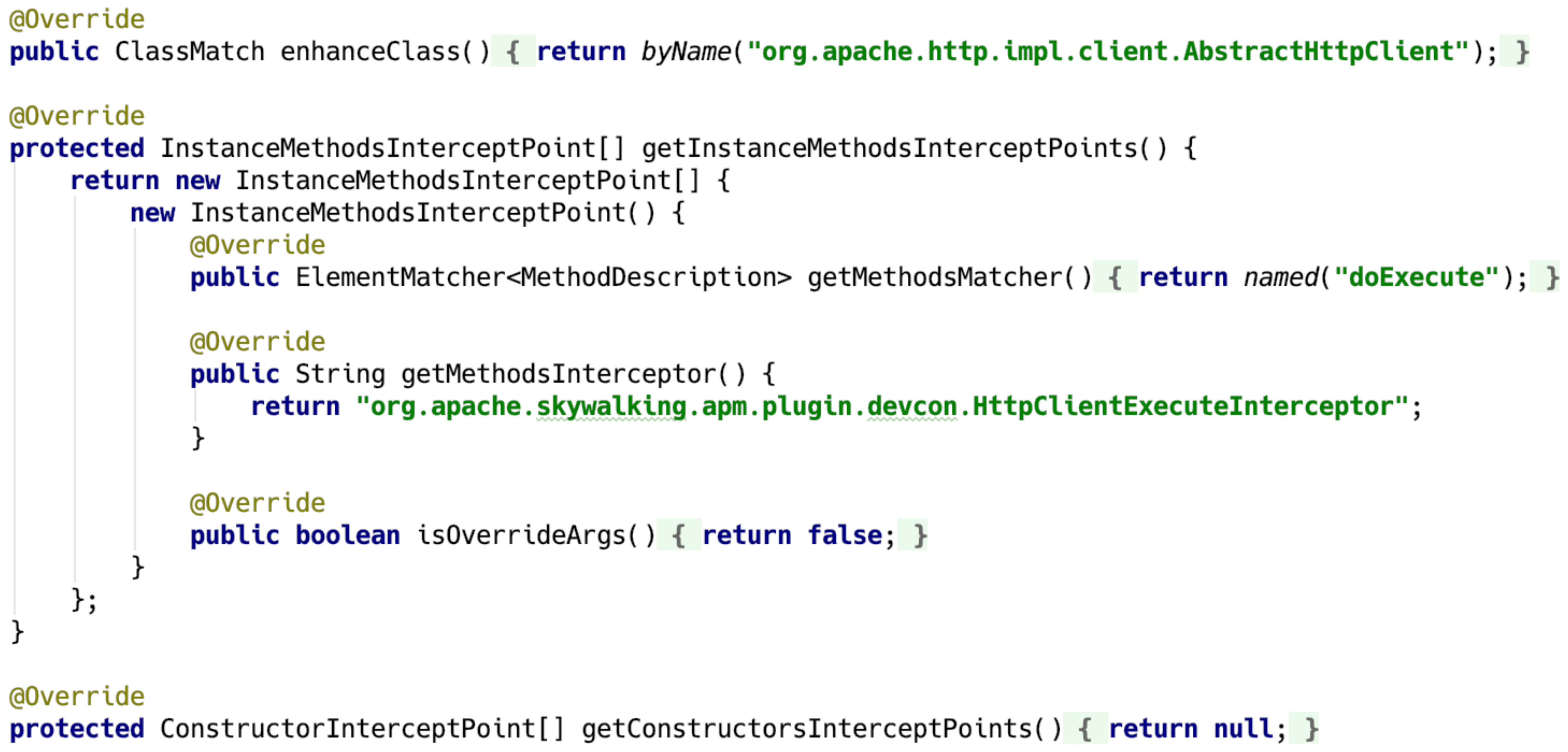
```java
@Override
protected ClassMatch enhanceClass() { return byName("org.apache.catalina.core.StandardHostValve"); }

@Override
protected ConstructorInterceptPoint[] getConstructorsInterceptPoints() { return null; }

@Override
protected InstanceMethodsInterceptPoint[] getInstanceMethodsInterceptPoints() {
    return new InstanceMethodsInterceptPoint[] {
        new InstanceMethodsInterceptPoint() {
            @Override
            public ElementMatcher<MethodDescription> getMethodsMatcher() { return named("invoke"); }

            @Override
            public String getMethodsInterceptor() {
                return "org.apache.skywalking.apm.plugin.devcon.TomcatInvokeInterceptor";
            }

            @Override
            public boolean isOverrideArgs() { return false; }
        }
    };
}
```

THE TOMCAT PLUGIN-INTERCEPTOR



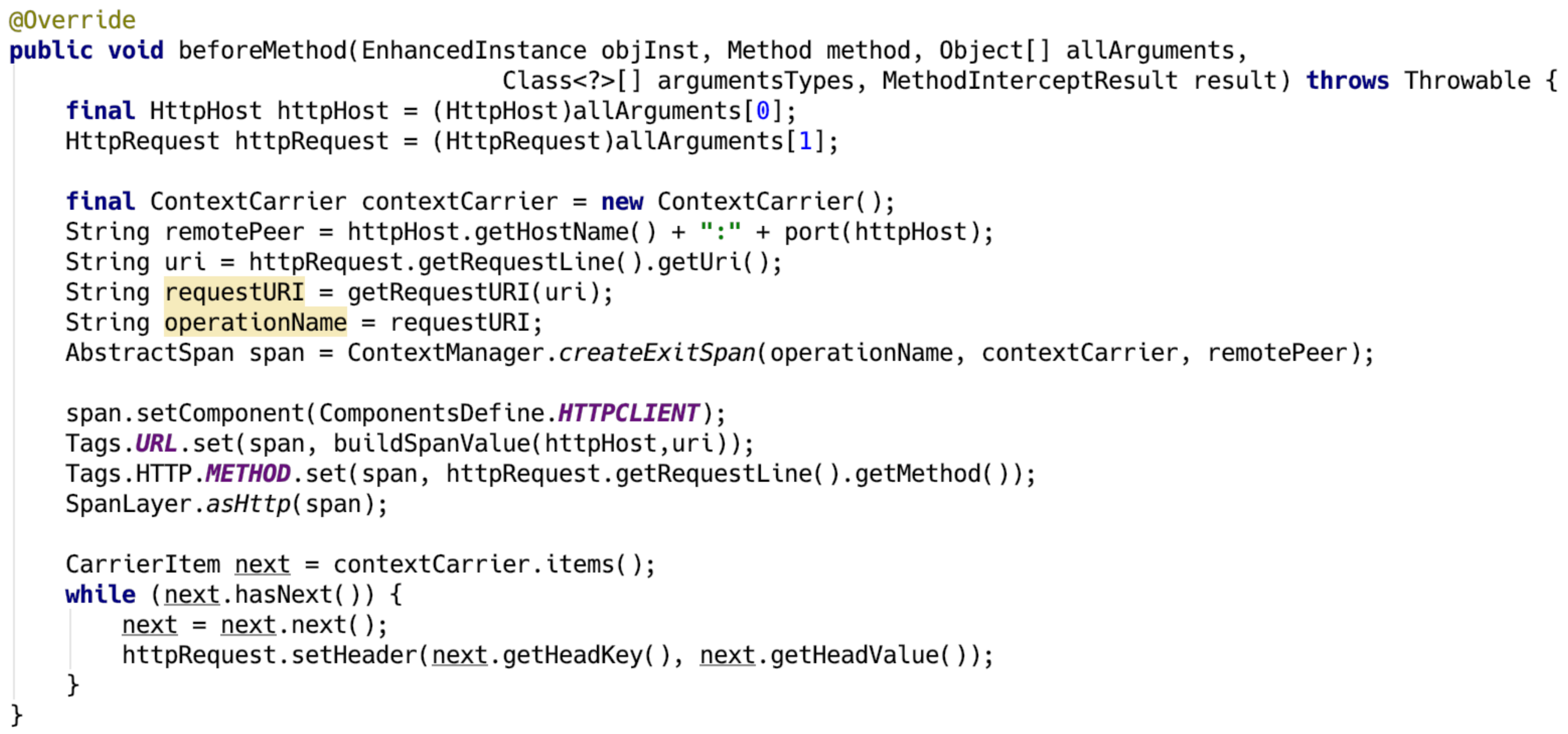
```java
@Override public void beforeMethod(EnhancedInstance objInst, Method method, Object[] allArguments,
    Class<?>[] argumentsTypes, MethodInterceptResult result) throws Throwable {
    HttpServletRequest request = (HttpServletRequest)allArguments[0];

    ContextCarrier contextCarrier = new ContextCarrier();
    CarrierItem next = contextCarrier.items();
    while (next.hasNext()) {
        next = next.next();
        next.setHeadValue(request.getHeader(next.getHeadKey()));
    }

    AbstractSpan span = ContextManager.createEntrySpan(request.getRequestURI(), contextCarrier);

    Tags.URL.set(span, request.getRequestURL().toString());
    Tags.HTTP.METHOD.set(span, request.getMethod());
    span.setComponent(ComponentsDefine.TOMCAT);
    SpanLayer.asHttp(span);

}
```

HTTPCLIENT PLUGIN - INSTRUMENTATION



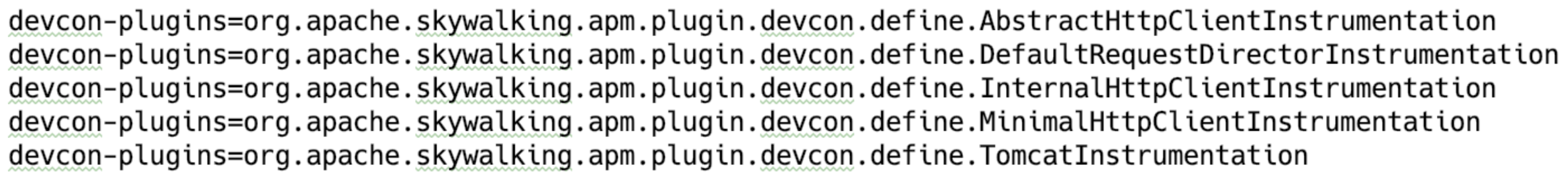
```java
@Override
public ClassMatch enhanceClass() { return byName("org.apache.http.impl.client.AbstractHttpClient"); }

@Override
protected InstanceMethodsInterceptPoint[] getInstanceMethodsInterceptPoints() {
    return new InstanceMethodsInterceptPoint[] {
        new InstanceMethodsInterceptPoint() {
            @Override
            public ElementMatcher<MethodDescription> getMethodsMatcher() { return named("doExecute"); }

            @Override
            public String getMethodsInterceptor() {
                return "org.apache.skywalking.apm.plugin.devcon.HttpClientExecuteInterceptor";
            }

            @Override
            public boolean isOverrideArgs() { return false; }
        }
    };
}

@Override
protected ConstructorInterceptPoint[] getConstructorsInterceptPoints() { return null; }
```

HTTPCLIENT PLUGIN - INTERCEPTOR



```java
@Override
public void beforeMethod(EnhancedInstance objInst, Method method, Object[] allArguments,
                                        Class<?>[] argumentsTypes, MethodInterceptResult result) throws Throwable {
    final HttpHost httpHost = (HttpHost)allArguments[0];
    HttpRequest httpRequest = (HttpRequest)allArguments[1];

    final ContextCarrier contextCarrier = new ContextCarrier();
    String remotePeer = httpHost.getHostName() + ":" + port(httpHost);
    String uri = httpRequest.getRequestLine().getUri();
    String requestURI = getRequestURI(uri);
    String operationName = requestURI;
    AbstractSpan span = ContextManager.createExitSpan(operationName, contextCarrier, remotePeer);

    span.setComponent(ComponentsDefine.HTTPCLIENT);
    Tags.URL.set(span, buildSpanValue(httpHost,uri));
    Tags.HTTP.METHOD.set(span, httpRequest.getRequestLine().getMethod());
    SpanLayer.asHttp(span);

    CarrierItem next = contextCarrier.items();
    while (next.hasNext()) {
        next = next.next();
        httpRequest.setHeader(next.getHeadKey(), next.getHeadValue());
    }
}
```

SKYWALKING-PLUGIN.DEF



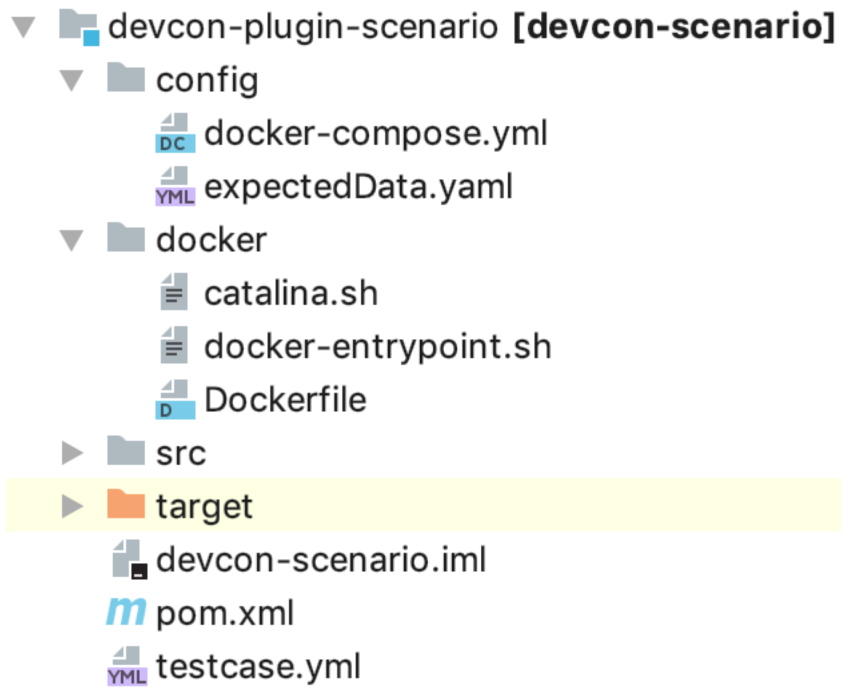
```
devcon-plugins=org.apache.skywalking.apm.plugin.devcon.define.AbstractHttpClientInstrumentation
devcon-plugins=org.apache.skywalking.apm.plugin.devcon.define.DefaultRequestDirectorInstrumentation
devcon-plugins=org.apache.skywalking.apm.plugin.devcon.define.InternalHttpClientInstrumentation
devcon-plugins=org.apache.skywalking.apm.plugin.devcon.define.MinimalHttpClientInstrumentation
devcon-plugins=org.apache.skywalking.apm.plugin.devcon.define.TomcatInstrumentation
```

TEST OUR PLUGIN

1. Design test case
2. Write test case project
    - Expected data file
3. Run the test case
    - Read the assert log
    - https://github.com/SkyAPMTest/agent-integration-testtool

SCENARIO PROJECT STRUCTURE

- expectedData.yaml
- testcase.yml
- docker-compose.yml
- docker/
- src



```
▼ devcon-plugin-scenario [devcon-scenario]
  ▼ config
      docker-compose.yml
      expectedData.yaml
  ▼ docker
      catalina.sh
      docker-entrypoint.sh
      Dockerfile
  ▶ src
  ▶ target
    devcon-scenario.iml
    pom.xml
    testcase.yml
```

RUNNING-COMMNAD

```
./deploy-test.sh --no-fetch_latest_code \
--no-clone_code \
--no-build \
--scenario devcon-plugin-scenario \
https://github.com/ascrutae/sky-walking.git \
feature/for-devcon
```

FQA

Thanks.

Zhang Xin, Skywalking PMC

2019/5/11